Teaching data structures through group based collaborative peer interactions

Nazir, Sajid; Naicken, Stephen; Paterson, James H.

Published in:
Proceedings of the 8th Computer Science Education Research Conference (CSERC '19)

DOI:
10.1145/3375258.3375270

Publication date:
2019

Document Version
Author accepted manuscript

Teaching Data Structures through Group Based Collaborative Peer Interactions

Sajid Nazir
sajid.nazir@gcu.ac.uk
Glasgow Caledonian University
Glasgow, UK

Stephen Naicken
s.naicken@alueducation.com
African Leadership University
Pamplemousses, Mauritius

James H. Paterson
james.paterson@gcu.ac.uk
Glasgow Caledonian University
Glasgow, UK

ABSTRACT

Data structures and algorithms is an important subject in Computer Science curriculum and builds upon the programming concepts learned by the students in their earlier courses. However, the abstract nature of the concepts can often be difficult for students to grasp. This problem becomes aggravated in an international setting with students from diverse academic backgrounds, resulting in some students losing interest and failing to follow along.

This paper describes our novel approach to teach data structures for Computing undergraduates from 30 African countries at a college in Mauritius in partnership with a UK university. The blended learning program uses as a student led "flipped classroom" approach, requiring students to view lecture and supporting material online prior to engaging in on-campus seminar session with the tutor.

Peer instruction is a key component of the flipped approach. In seminars, students worked on group based problem-solving activities in data structures supported by the tutor. The students devised their solutions on white boards taking ownership of the problem, became motivated to discuss their ideas freely, and to select a group solution. The group solutions were then shared with the other groups and peer reviewed, led by the tutor. This collaborative learning environment was observed to facilitate healthy discussions, and students' contributions and performance in later assessments offered evidence of understanding of core subject concepts.

CCS CONCEPTS

• **Social and professional topics** → **Computing education**.

KEYWORDS

data structures, group learning, active learning, blended learning

ACM Reference Format:
Sajid Nazir, Stephen Naicken, and James H. Paterson. 2019. Teaching Data Structures through Group Based Collaborative Peer Interactions. In *CSERC '19: The 8th Computer Science Education Research Conference, November 18–20, 2019, Larnaca, Cyprus.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Data structures and algorithms is an important subject in Computer Science curriculum [13] in universities throughout the world. Typically, the data structures course follows on from the introductory programming courses, and is considered useful for other advanced courses such as computer architecture, and networks etc. [4, 7]. Thus, it progresses students' knowledge required for understanding advanced programming concepts in their further courses [10]. The subject is also helpful for the students in their professional careers as software engineers, programmers, analysts, and information technology specialists.

Data structures concepts are abstract and the problems of its teaching stem from students failing to grasp these abstract concepts [13]. Low motivation of students is the first difficulty encountered by the tutors [12, 17]. Some students who seem to understand the concepts in class were unable to write programs [17]. Difficulties faced by the students in understanding basic data structures are identified in [19]. A common reason for students failing the data structures course is that they do not complete their programming tasks [9]. Thus, it is very important that the subject material and learning activities are engaging so that the students do not lose interest.

Visualizations and animations have found to be helpful in helping the students to understand the abstract concepts [7, 13]. Most students prefer a hands-on approach and enjoy the participation in the learning process [3]. It is imperative that the students are engaged in activities that motivate them to complete their programming assignments. The curriculum must also stress good programming style, which helps the students to write correct and efficient programs [10].

This paper describes the experience of teaching data structures and algorithms during 2017/18 in a bachelor's degree programme delivered at a newly established college in Mauritius. The mission of the college to bring together and train the future leaders of African countries, and the students were from a wide range different countries all across the African continent with diverse academic backgrounds. In keeping with this mission, the students are encouraged throughout the programme to be active learners through encouragement and participation in peer activities. The final degree is awarded by a UK university, and delivery is done as a collaboration between the college and the university, with online learning material initially developed in the UK and in-person teaching designed by tutors locally in Mauritius.

The in-person class sessions were key to the teaching approach. These involved peer discussion by students of problems set for them which led them to arrive at and articulate suitable solutions. The focus was on supporting the online learning content and activities through simple yet effective teaching innovations that made the subject interesting to the students. The experience showed that the students enjoyed the process, had developed better understanding of the core subject concepts. Healthy discussion ensued where the students could discuss and understand the relative merits of their group solution in relation to those developed by the other

students. The students became active learners and the results were remarkable.

The rest of the paper is structured as follows: Related work is described in Section 2. Methods and materials are covered in Section 3. Section 4 describes the relevant activities of group based peer interactions. The experiences and evaluations of the proposed method is provided in Section 5. Section 6 concludes the paper.

2 RELATED WORK

The use of competitive programming to teach data structures is reported in [4]. The students participated in a competition to improve their games project code against instructor-defined code and code of other students. Evaluation of code against other students was found to encourage students to put more efforts into their code. The project code was to be shared on a web server and the site ranked the students based on their code and other students were challenged to improve the ranking. The students were then evaluated based on code working properly.

The methodology of teaching data structures to be mathematical, theoretical or hands-on was considered in [8]. The three approaches were considered at different institutions by the panelists. The approach, where students were given problems and had to develop and analyze different possible solutions, helped produce a course with broad coverage preparing students to understand similar trade-offs in choice of data structures on their own. The students appreciated more hands-on approach during lab sessions. The students were assigned a major report writing exercise but then the danger is that the student would only be better at the data structure assigned. Thus, only a seminar/presentation style would be inadequate.

The use of different tools for teaching data structures and algorithms were explored in [3]. By actively involving the students, raised the students interest and helped the students to understand the various algorithms. The students' learning was improved with use of games and real-life examples. The teaching tools were used to supplement programming, and other assessments. Pieces of wood of varying lengths were used to teach about sorting algorithms with the students moving the wooden pieces according to the algorithm.

A technique termed as Visual Kinesthetic Psuedocode (VKP) was developed [13] with an aim to help the students code without coding, and providing necessary support in data structure implementation with actual code. VKP was designed to help the students to do actual coding by first helping the students to visualize the data structure, followed by textual pseudo-code that is, engaging the students in active learning.

The importance and use of online multimedia course content compared to face-to-face classroom teaching was explored in [7]. In order to increase the understanding of students they must be involved through active learning. Visualization through multimedia improves the learning experience of students.

Two tools for supporting the teaching were proposed in [1], visualizers provided visualizations of user data structures, whereas Testers then checked the implementation of visualizers. All the visualizers and testers were developed in Java. However, some students found it difficult to think of visualizers and testers as complementary tools.

Different techniques were used in [2] to make it easier to explain the data structure concepts to students. For example, simulations with software tools were used for teaching sorting, and Travelling Salesperson problem with play activity for students to visit all places to keep distance minimum.

For better learning of students, abstract concepts can be imparted using visualizations [14]. Thus animations can be developed to aid understanding of complex concepts such as Dijkstra algorithm. These animations can help the classroom teaching. Two training modes were used, with one in classroom to aid understanding, and the other to aid in programming.

Teaching of data structures in a creative way, Creative Lab with Active Participation (CLAP), is described in [12]. This provided an open problem solving approach, where solutions were articulated using an algorithm animation environment . The teaching is student-centric and the students had to take initiative. The CLAP course was conducted in student pairs.

An active learning approach for teaching data structures to postgraduate students is described in [15]. The aim was to increase the programming concepts learning in the lab. All exercises were open ended and the students could use their own ways and means to solve it. This encouraged students to come up with different applications from other courses being studied.

Interactive learning through visual environments for automatic feedback to students was used in [11]. Students also provided feedback to each other by peer reviewing and it was observed to produce good learning results. Use of graphics to teach algorithms and data structures was explored in [5]. Image processing and rendering projects were used to teach memory allocation and matrix manipulation. Computer graphics provided a mechanism to teach problem based learning.

Much of the above work is based on tools and visualizations to support learning. In contrast, our approach is focused on the active learning that takes place through group based peer interactions and collaborations in a problem-solving context.

3 METHODS AND MATERIALS

3.1 Learning environment

The data structures course was taught to the first cohort of students in July 2018. The course was delivered using a blended learning approach. To support the approach, online material was developed to present the theory of a range of data structures and examples of their implementation and use. The course materials were made available through a Virtual Learning Environment (VLE) and students were expected to study the online material before coming to the class. In the active learning class session, the instructor presented a real-world problem that could be solved using data structures which had been introduced in the online material. In class, concepts were reinforced and further explored through peer interactions both within and between groups, and finally the tutor summarized the discussion by providing feedback on the strengths and weaknesses of the various group approaches.

The students were from a wide range of different countries and have experienced widely differing teaching methodologies and first programming languages in their previous education. However, they have on their current programme become accustomed to a blended

Trimester	Module	Language
Year 1	Programming 1 Programming 2	Java Java
Year 2	Data Structures and Algorithms	Java
Year 3	Big Data	Python

Table 1: Programming related courses on the programme

approach and highly interactive peer learning activities in class. The challenge in this particular course was to apply this approach and exploit the students' collaborative skills in motivating and facilitating learning abstract concepts which often do not engage the interest of students.

3.2 Choice of language and data structure

The concepts taught during a typical computing science data structures curriculum are quite advanced. The programming related courses at the African college are shown in Table 1.

Students are exposed to Java during year 1. Java is a simple and elegant programming language for teaching data structures compared to C/C++. The use of frameworks such as Standard template Library (STL) is described in [16] which is important to teach the integration of user code with existing libraries. Teaching of data structures with Java language is described in [18], highlighting the improvements that the language has over C++.

The advantage of using Java for the students was that they were already familiar with the core language concepts from the earlier programming modules. Hence they can concentrate on understanding and implementing the data structure constructs without being distracted or hindered by the language elements.

Data Structures curriculum was taught by reinforcing the theoretical concepts with innovative practical activities. The initial discussion about a data structure was used to relate the abstract concepts to real world usage of those concepts known to be familiar and of interest to the students. For example, for introducing the students to trees, students were given examples of Document Object Model (DOM) with which they were familiar by studying HTML and XML document structure from the web platform development module.

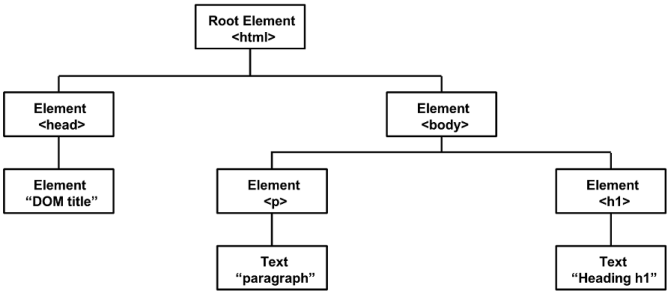
Figure 1 shows the DOM representation as a tree. As can be seen the document has nodes that correspond to the tree structure.

In order to highlight our strategy for improving the student experience by engaging them in group peer interactions, we have chosen the DOM based problem that was used to teach the tree data structure. The problem assigned to the students was to develop a pseudocode solution and Java code for populating a tree data structure by reading data from a given HTML document that uses a simple subset of HTML.

3.3 Open plan environment

It is important for free peer interactions that the classroom activities are flexibly designed to encourage open discussions between the group members. We used the traditional white board approach for sharing the group's work in progress by assigning each group to one of the whiteboards that were installed along the walls all

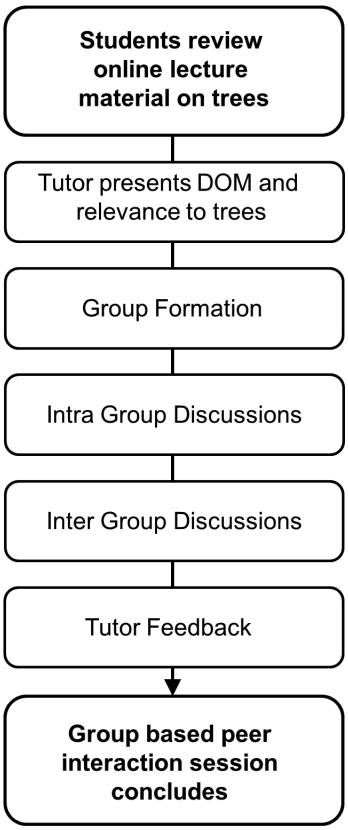
Figure 1: Document object model displayed as a tree.



around the classroom. This physcial environment does not rely on any expensive technology, but it is a simple and effectively designed space.

This allowed for peer collaboration within a group without interfering with the discussions or disturbing the other groups. Also, it made it easier for the tutor to coordinate and interact with the group activities. The setting enabled the students in later stage of inter group discussion to look at the reasoning, approach and development of solutions by other groups.

Figure 2: Sequence of actions to communicate full understanding of a data structure.



3.4 Group formation

The idea of group is central to our approach as the peer interactions take place both within and between groups. It was therefore critical to make a different group for the next problem. The groups were formed by the tutor by assigning students randomly to different groups. This helped students not only to work with different peers and hence with students of varying backgrounds but also helped them learn the group dynamics and peer support. Peer interactions helped to reinforce the data structures and algorithms concepts being studied providing an active learning environment for the students.

3.5 Facilitated learning

The sequence of actions with the learning activity is summarized in Figure 2. The problem to be solved was introduced to the students before they were divided into groups. This sets a baseline for all the groups, that is, for the DOM Tree scenario they were introduced to the construction of the DOM as a real-world problem that motivates the use of a data structure. The groups were required to fully develop a solution in a top-down manner, initially within the group, and are encouraged by the tutor to also capture their solution through diagrams, flow charts and pseudo code on the white-board. Students do not use their laptops when working on the problem, so that the focus is not on code and implementation, but the design of the algorithm and appropriate use of the relevant data structures, so that students can develop their computational thinking. The tools used in the session aid and enhance the student learning and experience [10]. This also makes it possible for the tutor to see the progress and the thought process of the group. This is helpful to the tutor to later reinforce the correct approaches taken by different groups highlighting the differences and merits in each case.

4 GROUP PEER INTERACTIONS

In this section, we describe the group activities and show how these contributed to improve the students' understanding through peer interactions.

The interactions were closely watched by the tutor and students were prompted and encouraged to analyze their solutions as they were being developed. The interactions took place within a group to arrive at a mutually agreed solution to the given problem.

This also helped the tutor to encourage and direct the students in arriving at the class wide best solution and see for themselves the other possible solutions together with their strengths and weaknesses.

4.1 Student intra-group interactions

The design of the solution by the students required three distinct stages, (i) an initial informal discussion in the group with charting the program elements on the white board, (ii) iterative refinement of the solution with peer interaction which is also observed by the tutor, (iii) development of a fully refined solution. Figure 3 shows the free-form scribbling on the board for one of the groups to design and develop a group solution.

The initial discussions in the group sometimes resulted in emergence of one or more group leaders due to group dynamics where each member was trying to increase their influence. This was beneficial for the group because in peer learning they can quickly follow a leader to start developing their solution through distribution of work and coordination of effort.

Figure 3: Output from group discussions on whiteboards

The students were encouraged to use flowcharts and pseudo-code to help outline, develop and discuss their solutions. This approach of a free form solution design had benefits similar to VTK [13] by helping students to understand the core tree data structure before implementing it in Java code.

The solutions were developed on the whiteboard and students were encouraged to graphically depict the solution as shown in Figure 3. This also made it easier for the tutor to move around between groups and discuss their progress and provide feedback to put them on a desired course of action. The whiteboards as an aid helped significantly as the tutor could easily scan the progress and direction of each group from a central position. The individual participation and leadership of each group also became apparent to the tutor who can then encourage those students who were not fully engaged with the group activities. The students in general seemed to be very keen to learn through peer interactions and took ownership of the learning process.

The algorithm efficiency was also considered during the solution development process as ultimately it decided how useful a solution really was in terms of time and space complexity. The emphasis was on developing efficient solutions [10]. It was critical for students to complete their assignments up to a good standard [9].

The development of a working solution was a good starting point and a motivator for the students to work hard to improve their solution. After that the tutor encouraged the groups to have a deep look at their solutions and see how those could be made better. The in-group discussions and attempts to improve the group solution ensured that every group and student got to understand the weaknesses and strengths of various possible approaches to solving the given problem.

4.2 Student inter-group interactions

During inter group deliberations, the students took turns in discussing their group solution with other groups. The tutor assumed the role more like a moderator to keep the discussion on course. The objective was to make it easier for the students to arrive at the best solution to a given problem and how their group solution compares with other solutions. The students were found to be very excited and keenly took part to logically give arguments to support their approach and solution. The student understanding of the topic was helped by the fact that they were excited to ask questions and provide answers, displaying a scientific approach, positive criticism and feedback.

The process comprised of the following steps:

4.2.1 Presentation of solution to other groups. After the students in all the groups have had intra-group discussions and converged to a solution, then the groups were asked to present their solution and approach to the other groups. Other groups asked questions with an aim to understand the approach taken and to uncover any perceived weaknesses in the solution.

4.2.2 Peer Evaluation. Peer evaluation made other groups familiar with the data structure under consideration and thus students could clearly see the advantages and limitations of various approaches together with their time and space complexities.

4.3 Round-up Discussion

After all groups had presented their solution, the tutor described the relative merits of each approach and concluded by highlighting the best approach. Students had the opportunity to ask questions to the facilitator and any group members to further understand the proposed solutions or the facilitator's prefered approach.

At the end of the session, the tutor requests that students complete the implementation of the problem prior to the next session. Students share their code on Github Gist[1] where the facilitator and other students may leave feedback using the comments system. Students are encouraged to leave peer feedback and to iteratively improve their implementation with respect to feedback.

4.4 Benefits

The students perceived both the intra and inter group interactions to be of great benefit to their understanding of data structure concepts. The group interactions provided a means for the students to learn in a fun way. The student involvement in many practical exercises helps to reinforce the concepts and develops the skill set of the students [6].

The active learning environment in the group setting made the students more confident to voice and share their ideas. In some groups it was noticed that one or two peer teachers emerged that were leading the discussion. However it was found to be of benefit to all students.

The group interactions and discussion around various implementations of the same problem is important as this comparison and evaluation of the different solutions fosters better understanding of the critical concepts which are required by the student in future career as programmer, analyst etc. [10].

[1] http://gist.github.com

5 EVALUATION

The feedback received for the proposed technique was very positive and some students described Data Structures and Algorithms module as the best part of the programme and that it helped them understand the difficult concepts. The technique of collaborative peer learning was tested on a class of 37 students with varying programming backgrounds and skills. The interest of students in the module content was found to have increased considerably and the students rated their learning environment of data structures to be the best compared to other modules that were using only a flipped classroom approach.

5.0.1 Student Feedback. The feedback received for the proposed technique was very positive and some students described it as the best part of the study programme and that it helped them demystify and understand the difficult core module concepts.

A prominent theme in student feedback was that the students benefited from the experiential learning methodology – "Experiential learning, made it difficult to forget the learning. We learn better by doing". On the use of videos showing dancers implementing search algorithms, which were then implemented in code by students, one student stated, "It facilitated out of class learning. I saw the search algorithm happen, it looked so simple, so I felt encouraged to look at the algorithm and investigate further, and question it."

5.0.2 Inspiration. Many students reported being really inspired as a result of taking ownership of the learning process through group based peer interactions. They reported that their initial barriers and fears of learning a difficult topic such as trees were adequately overcome.

5.0.3 Student experience. The availability of the online material and the expectation that they had to study it before coming to the classroom improved the student experience. The tutor led classroom activities reinforced the core concepts with the group interactions enabling the students to learn actively with their peers. The students who would otherwise not participate in classroom activities taking the whole classroom as a group, also feel empowered and contribute to the group discussion thus enhancing their learning in the process. The ultimate result is an enjoyable learning experience for the students.

5.0.4 Improvement in results. The module assessment comprised of both theoretical knowledge and concepts, and practical programming of data structures. The pass rates for the module were better than other modules where the techniques of group based peer interactions were not applied.

6 CONCLUSION AND FUTURE WORK

This paper has described the trialing of an innovative technique of creating an easy to access discussion medium to peer learn and teach an important and complex tree data structure concept in a classroom environment. The peer interactions made it possible to freely discuss their ideas and approaches to solve the given problem. The students could easily articulate their solution on the whiteboards readily available to the instructor and peers to see, comment and improve upon.

The objective was to make it easier for the students to interact and understand the complex and abstract data structure concepts through active learning. The interactions were facilitated by dividing the class into random equal-sized groups and encouraging them to devise a group solution to the given problem by making use of white board to freely share and comment ideas. The group solutions were then shared across groups and invited comments in terms of completeness and efficiency from other groups. It was observed that the students acquired thorough understanding of the data structure under study and felt empowered and confident to tackle complex problems in data structure curriculum.

In our future work we aim to improve the benefits of this study further by announcing the group problem before the actual class so that the students get more time and a chance to devise their individual solutions and before coming to the class. We would also consider to record the understanding of students of a particular data structure both before and after group interactions in order to quantify the benefits achieved through peer learning.

Empowering the students with the right training of the data structures prepares them well for the diverse and complicated data structures that they are most likely to encounter in real-world settings. [10].

REFERENCES

[1] Ryan S Baker, Michael Boilen, Michael T Goodrich, Roberto Tamassia, and B Aaron Stibel. 1999. Testers and visualizers for teaching data structures. *ACM SIGCSE Bulletin* 31, 1 (1999), 261–265.

[2] Suhas Bhagate and Uday Nuli. 2016. Innovative Methods for Teaching Data Structures and Algorithms. *Journal of Engineering Education Transformations* (2016).

[3] Martin J Biernat. 1993. Teaching tools for data structures and algorithms. *ACM SIGCSE Bulletin* 25, 4 (1993), 9–12.

[4] Donald Chinn, Phil Prins, and Josh Tenenberg. 2003. The role of the data structures course in the computing curriculum. *Journal of Computing Sciences in Colleges* 19, 2 (2003), 91–93.

[5] Andrew T Duchowski and Timothy A Davis. 2007. Teaching algorithms and data structures through graphics. In *Proc. of Eurographics*.

[6] Alan Fekete. 2002. Teaching data structures with multiple collection class libraries. In *ACM SIGCSE Bulletin*, Vol. 34. ACM, 396–400.

[7] Sahalu Junaidu. 2008. Effectiveness of multimedia in learning and teaching data structures online. *Turkish Online Journal of Distance Education* 9, 4 (2008), 97–107.

[8] Danny Kopec, Richard Close, and Jim Aman. 1999. How should data structures and algorithms be taught. In *ACM SIGCSE Bulletin*, Vol. 31. ACM, 175–176.

[9] Ramon Lawrence. 2004. Teaching data structures using competitive games. *IEEE Transactions on Education* 47, 4 (2004), 459–466.

[10] Karen Mackey and Howard Fosdick. 1979. An applied computer science/systems programming approach to teaching data structures. In *ACM SIGCSE Bulletin*, Vol. 11. ACM, 76–78.

[11] Lauri Malmi and Ari Korhonen. [n. d.]. A pedagogical approach for teaching data structures and algorithms.

[12] Veijo Meisalo, Erkki Sutinen, and Jorma Tarhio. 1997. CLAP: teaching data structures in a creative way. In *ACM SIGCSE Bulletin*, Vol. 29. ACM, 117–119.

[13] Ogen Odisho, Mark Aziz, and Nasser Giacaman. 2016. Teaching and learning data structure concepts via Visual Kinesthetic Pseudocode with the aid of a constructively aligned app. *Computer Applications in Engineering Education* 24, 6 (2016), 926–933.

[14] Deng Rui, John T Thompson, Yang Hong, Zhou Xing-sheng, Liu Ke-jing, and Neil Alexander Macintyre. 2008. Imagery training in the teaching of the data structure curriculum. *ACM SIGCSE Bulletin* 40, 4 (2008), 92–94.

[15] C Sujatha, GN Jayalaxmi, and GK Suvarna. 2012. An innovative approach carried out in data structures and algorithms lab. In *2012 IEEE International Conference on Engineering Education: Innovative Practices and Future Trends (AICERA)*. IEEE, 1–4.

[16] Josh Tenenberg. 2003. A framework approach to teaching data structures. In *ACM SIGCSE Bulletin*, Vol. 35. ACM, 210–214.

[17] Zhao Wang. 2012. The Research on Teaching Ideas of" Data Structure and Algorithm" in Non-computer Major. In *Advances in Computer Science and Education*. Springer, 249–254.

[18] Mark Allen Weiss. 1997. Experiences teaching data structures with Java. *ACM SIGCSE Bulletin* 29, 1 (1997), 164–168.

[19] Daniel Zingaro, Cynthia Taylor, Leo Porter, Michael Clancy, Cynthia Lee, Soohyun Nam Liao, and Kevin C Webb. 2018. Identifying Student Difficulties with Basic Data Structures. In *Proceedings of the 2018 ACM Conference on International Computing Education Research*. ACM, 169–177.